International Journal of Modern Physics C
Vol. 20, No. 2 (2009) 295–311

World Scientific
www.worldscientific.com

GEOMETRIC RANDOM INNER PRODUCT TEST AND RANDOMNESS OF π

FATIN SEZGIN

Department of Business Information Management
Bilkent University, Ankara 06800, Turkey
fatin@bilkent.edu.tr

Received 23 March 2008
Accepted 24 September 2008

The Geometric Random Inner Product (GRIP) is a recently developed test method for randomness. As a relatively new method, its properties, weaknesses, and strengths are not well documented. In this paper, we provide a rigorous discussion of what the GRIP test measures, and point out specific classes of defects that it is able to diagnose. Our findings show that the GRIP test successfully detects series that have regularities in their first- or second-order differences, such as the Weyl and nested Weyl sequences. We compare and contrast the GRIP test to some of the existing conventional methods and show that it is particularly successful in diagnosing deficient random number generators with bad lattice structures and short periods. We also present an application of the GRIP test to the decimal digits of π.

Keywords: Geometric random inner products; random number generator; randomness testing; randomness of π; Monte Carlo; pseudorandom.

PACS Nos.: 02.50.Cw, 02.50.Ng, 02.70Uu, 05.10.Ln.

1. Introduction

Random number generators became the focus of attention in the last decades because of widespread use of simulation studies enjoying vast possibilities provided by computers. As in every branch of science, Physics is also employing Monte Carlo as a reliable research methodology in applications such as quantum particle scattering, Ising spin, percolation, or random walk models. This technique is used as a tool for theory building, model development, input data generation, and hypothesis testing. The operating model obtained by simulation must imitate closely the random behavior of a real system in order to obtain reliable and valid inferences. Users must be very careful because several generators considered as having good quality and receiving universal use are later proved to be seriously flawed and unusable. Random number generators relying on a mathematical formula generate deterministic sequences that resemble to a random output from a truly random process. A reliable random number source must pass as many theoretical and empirical tests

as possible, especially tests related to properties required in a particular target application. For this reason, discovery of new tests is always required and appreciated by the simulation community. In their recent papers,[1,3] Tu and Fischbach presented a newly developed test and called it Geometric Random Inner Product (GRIP). In a previous paper,[4] they investigate the randomness of digits of π by using this test. Although this work supports the opinion of previous authors[5,8] for the appropriateness of π as a random number generating source, finds it inferior to some other generators. The objectives of our study are to review the GRIP as an empirical test tool, and to point out some weaknesses and suggest some improvements. For this purpose, the test is discussed briefly in Sec. 2 and the method of calculation for its exact mean and variance is demonstrated. The appropriate significance test procedure is explained and shortcomings of present applications are pointed out. In Sec. 3, several important points are noted about the previous work of authors on the randomness of π. The success of multiplicative combining technique is demonstrated by implementing the method correctly and valid results are calculated. Section 4 presents some additional remarks.

2. The GRIP Test

GRIP is a family of tests that use the expected value of $\langle \mathbf{r}_{12} \cdot \mathbf{r}_{23} \rangle_n$ or $\langle (\mathbf{r}_{12} \cdot \mathbf{r}_{23}) \cdots (\mathbf{r}_{2m-1_{2m}} \cdot \mathbf{r}_{2m1}) \rangle_n$ as parameters where $\mathbf{r}_{ij} \cdot \mathbf{r}_{jk}$ denotes the inner product of two vectors. In this notation, $\mathbf{r}_{ij} = \mathbf{r}_j - \mathbf{r}_i$ values are difference of coordinates $\mathbf{r}_j$ and $\mathbf{r}_i$ in n dimensional object, and $2m$ is a positive even integer representing the number of uniform points configurations. For example, for the case of three-dimensional space and $m = 2$, we can obtain the following set of vectors:

$$\mathbf{r}_1 = \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix}, \quad \mathbf{r}_2 = \begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix}, \quad \mathbf{r}_3 = \begin{pmatrix} x_3 \\ y_3 \\ z_3 \end{pmatrix}, \quad \mathbf{r}_4 = \begin{pmatrix} x_4 \\ y_4 \\ z_4 \end{pmatrix}, \tag{1}$$

$$\mathbf{r}_{12} = \begin{pmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \end{pmatrix}, \quad \mathbf{r}_{23} = \begin{pmatrix} x_3 - x_2 \\ y_3 - y_2 \\ z_3 - z_2 \end{pmatrix},$$

$$\mathbf{r}_{34} = \begin{pmatrix} x_4 - x_3 \\ y_4 - y_3 \\ z_4 - z_3 \end{pmatrix}, \quad \mathbf{r}_{41} = \begin{pmatrix} x_1 - x_4 \\ y_1 - y_4 \\ z_1 - z_4 \end{pmatrix}. \tag{2}$$

Here x_i, y_i, and z_i are the coordinates of the ith random point. Tu and Fischbach obtain these coordinates by expressing several consecutive digits of the output of random number generator as a uniform number between -1 and $+1$. We present a Fortran 77 program listing of GRIP test at Appendix A for three and four dimensions each having three points.

2.1. *The parameters of test statistics*

In their earlier paper,[2] Tu and Fischbach ranked entries in terms of their errors expressed as absolute values between expected and computed results. This causes the omission of standard deviations. Without any standardization or significance test, authors rank generators with respect to their absolute deviations and conclude that all generators except nested Weyl sequence and Weyl perform better in $n = 3$ than $n = 9$. In later studies, the deviations are expressed in standard errors. However, we must point out that in the conventional statistical notation it is not appropriate to denote these sample statistics with σ. Because these standard errors (not standard deviations) are calculated from the samples and are not parametric values. In some cases, the usage of statistics instead of parameters may cause serious mistakes in comparing various generators.

A reliable comparison must use test statistics and employ their distribution theory in order to reach objective and sound inferences. When possible, existence of parametric values for estimators and their variances is preferable. For this reason, we briefly present derivation of necessary parametric values. Formula (3) presented by Tu and Fischbach[4] for finding expected values of inner products in n-cubes is very complicated. A simpler formulation can be obtained by using properties of expectation operator E and standard uniform distribution. Consider, for example, $z = (x_2 - x_1)(x_3 - x_2) = 4(u_2 - u_1)(u_3 - u_2)$. Since u_i are independent uniform (0, 1) random variables,

$$\frac{E(z)}{4} = E(u_2u_3 - u_2^2 - u_1u_3 + u_1u_2) = E(u_2u_3) - E(u_2^2) - E(u_1u_3) + E(u_1u_2)\,. \quad (3)$$

By independence, we can write $E(u_iu_j) = E(u_i)E(u_j) = 1/2\ 1/2 = 1/4$. Since

$$E(u^k) = \int_0^1 u^k du = \frac{1}{k+1} \quad (4)$$

in uniform (0, 1) distribution, we get $E(z) = -1/3$. Therefore, since $\langle \mathbf{r}_{12} \cdot \mathbf{r}_{23} \rangle_n$ is the sum of n independent z values of the above form, it has the expected value $E\langle \mathbf{r}_{12} \cdot \mathbf{r}_{23} \rangle_n = -n/3$. Expected values of more complicated terms can be found by noting that expected value of independent uniform variables with various powers forming a multiplication can be calculated as

$$E(u_1^{k_1}u_2^{k_2}\cdots u_m^{k_m}) = E(u_1^{k_1})E(u_2^{k_2})\cdots E(u_m^{k_m}) = \frac{1}{k_1+1}\frac{1}{k_2+1}\cdots\frac{1}{k_m+1}\,. \quad (5)$$

By using this approach, we have calculated the parametric values of means and variances presented in Tables 1 and 2 of Tu and Fischbach.[4] The results are summarized below:

Form	Mean	Variance
$\langle \mathbf{r}_{12} \cdot \mathbf{r}_{23} \rangle_3$	-1	38/30
$\langle \mathbf{r}_{12} \cdot \mathbf{r}_{23} \rangle_6$	-2	38/15
$\langle (\mathbf{r}_{12} \cdot \mathbf{r}_{23})(\mathbf{r}_{34} \cdot \mathbf{r}_{41}) \rangle_3$	4/3	4144/675
$\langle (\mathbf{r}_{12} \cdot \mathbf{r}_{23})(\mathbf{r}_{34} \cdot \mathbf{r}_{41}) \rangle_6$	14/3	24 764/675

These parameters can be calculated also by employing high-level interpreted programming languages of computer algebra systems. It would be safer to use these values for error grading or hypothesis testing purposes. Otherwise, because the underlying distribution may not be uniform (0, 1), using variances calculated from the data may cause incorrect results. From the table values of errors in Tu and Fischbach,[4] we infer that, for example, in their Table 1, the sample standard error of NSW is 0.00206, whereas it is 0.00159 for most entries. In Table 2 for $n = 3$, the standard error is about 0.00248, but NWS has a value of almost double: 0.00471. For $n = 6$, NWS gives a slightly higher value: 0.00813 instead of 0.00606. In Table 3, $n = 3$ the standard error of NWS is more than triple of other generators, whereas it is smaller than others for $n = 6$. In Table 6 the standard errors of multiplicative combinations are almost half of other entries.

Other important information obtained from variances is the comparison of relative magnitudes of sample statistics in different generators. In addition to means, we can also test the variance estimates. The sample variance S^2 has a mean value of σ^2 and variance $(\mu_4 - \mu_2^2)/n$ in large samples.[9] Here μ_4 and μ_2 are fourth and second expected moments about the distribution mean. In Sec. 3.1, since n values are very large, by employing central limit theorem for the distribution of sample variance we used this method as an additional evidence for the failure of some generators in GRIP test or indication of incorrect implementations.

2.2. *Significance level versus ranking*

The usual approach for testing randomness by empirical statistical tests is to use significance levels instead of letter grading. Obtaining A^+ from all tests in a battery may seem desirable at the first glance but it is another indication of lack of randomness called "too good fit." For example, uniformity tests of random number generators will give χ^2 values very close to zero if the size of the tested data approaches the cycle length of the generator. For this reason, too good fit can be considered as a symptom of nonrandomness. Taking this fact into consideration, Knuth[10] suggests the following indications for the significance levels:

0–1%, 99–100%, Reject randomness;
1–5%, 95–99%, Suspect randomness;
5–10%, 90–95%, Almost suspect randomness.

Therefore for a reliable testing procedure, we must not consider only extreme deviations, but also very small deviations. A better practice is to apply a test several

times by dividing the data into smaller disjoint sets and convert test statistics to p-values. By employing two-level tests, the uniformity of these values can be assessed by empirical distribution function goodness of fit techniques[11] such as Anderson–Darling, Kolmogorov–Smirnov or Cramer–von Mises.

Adopting this approach, instead of unusual grading system of results, Marsaglia[12] applied two tests of randomness. First, he obtained the exact mean and variance for the product of the form: $(x_2 - x_1)(x_3 - x_2) = 4(u_2 - u_1)(u_3 - u_2)$. Anderson–Darling test for uniformity conducted on 32 mean values each consuming consecutive 30 million decimal digits of π supported the randomness hypothesis. Second, using a more rigorous application of GRIP test, he obtained the exact distribution function of $Z = (u_2 - u_1)(u_2 - u_3)$ as

$$F(z) = \frac{1}{3}\left\{(1-8z)\sqrt{1+4z} + 6z\ln\left(\frac{1+\sqrt{1+4z}}{1-\sqrt{1+4z}}\right)\right\} \quad \text{for } -\frac{1}{4} < z < 0\,,$$

$$= \frac{1}{3} + \frac{2}{3}\{4z^{3/2} - 3z\ln(z) - 3z\} \qquad \text{for } 0 \le z \le 1\,.$$

and by using again Anderson–Darling test on 32 values each obtained from 30 million consecutive decimal digits, he concluded that the distribution function also supports the suitability of the expansion of π as a source of independent random variables.

3. Application to π and Generalizations

It is known that π is normal; therefore, every sequence of n digits is equally likely to occur. However, although this fact is used by several authors as an indication of the global randomness of π, it does not imply local randomness for particular usages. In fact, there are remarkable patterns in decimal expansion of π as quoted by Knuth.[10] This is a dilemma frequently envisaged in generating and testing random numbers. Empirical statistical tests can reveal local nonrandomness in a sequence, but surviving a battery of tests cannot prove its randomness. Moreover, empirical tests are not finite and new methods can always be devised as in the GRIP family of tests. Therefore, statisticians tend to rely more on theoretical properties of a random number generator. Two of these theoretical properties deserve a special attention. One of them is the serial correlation that measures the correlation between pairs of terms X_i and X_{i+k} produced k units apart. Other theoretical property investigated for random number generators is expressed by figures of merit representing the discrepancy and spectral test results applicable in certain generator classes. In congruential random number generators, t-dimensional vectors of successive numbers in dimension $t \ge 2$ have lattice structure. Generated points fall on parallel hyperplanes and properties of these lattices can be used as theoretical quality measurements called figures of merits for comparing various generators. For example, normalized distances $S_t = d_t^*/d_t$ is a common quality measure comparing d_t, the maximum distance between adjacent hyperplanes determined by the points

of the lattice in t-dimensional space and d_t^*, the lower bound of this distance. Actually, a purportedly random sequence obtained by a deterministic formula cannot be random in the classical sense of the word. Many authors writing on RNGs quote the famous expression of John von Neumann: "Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin." The random number generators used in practice are not only deterministic but also periodic. Irrational and transcendental numbers, on the other hand, are not periodic but some of them can be expressed and produced by a finite formula. According to Kolmogorov's definition, an infinite sequence of bits is random if it cannot be described by a sequence shorter than itself. Now we have an efficient formula to calculate any arbitrary bit of π expansion without having to find intermediate values. Therefore, randomness tests are essential before using particular subsequences of π as a random number source for a target application. In this respect, the efforts of Tu and Fischbach[4] are noteworthy. However, readers must be reminded that their study covers only a very small section at the beginning of π and results cannot be generalized to other segments of this irrational number with infinitely many decimal digits. As Marsaglia[12] points out, the study of Tu and Fischbach produced a worldwide interest and over 400 internet sources created the impression among nonstatisticians that π gets a poorer randomness grade compared with some other sources. In his criticism, Marsaglia tested 360 million decimal digits of π and showed that GRIP does not provide any significant indication for a weakness of randomness of π.

3.1. *Combining π with other generators*

Combining is a very effective way of improving random number generators. In testing the first 10 000 digits of π, in blocks of 1000 digits each, Pathria[5] discovered certain nonrandom blocks. Pathria remarks that these patterns are dangerous even if diluted by one of their neighboring blocks. Therefore, combining random numbers from several different sources is seen by many authors as a remedy to iron-out irregularities of a single generator. Tu and Fischbach present related test results in their Tables 5 and 6. The success of combining by π in improving the quality of outputs is obvious for several instances. However, there is a peculiar situation for combined generators obtained by multiplication. Although combinations obtained using $(+)$ and $(-)$ operations exhibit very satisfactory results, all cases containing $(\times)$ operation give almost half of the expected values and huge errors. This situation arises from an incorrect application in multiplying random numbers. Legitimate functions for improving the quality of pseudorandom numbers are presented by Deng and George.[13] Their work covers independent random variables with continuous density function on the interval (0, 1). It will be useful to try these functions during the combination effort. Unfortunately, multiplying uniform (0, 1) random variables does not produce uniform output. This distribution is given in many sources[14] as $f(z) = -\ln(z)$ and can be easily derived by considering the pdf of the random variable obtained by multiplying two uniform random variables.

When X and Y have uniform distribution in the (0, 1) interval, the distribution function of $Z = XY$ can be written as

$$F(z) = 1 - \int_z^1 dx \int_{z/x}^1 dy = z(1 - \ln(z)) . \tag{6}$$

Therefore, by differentiation one can obtain the pdf as

$$f(z) = -\ln(z) . \tag{7}$$

This fact explains the inconsistency of outcomes for the multiplicative cases. In random number generators, several authors have used multiplication but this was applied for integers and subjected later to a Mod operation. For example, the so-called power generator discussed by Lagarias[15] uses the recursion

$$x_{n+1} = x_n^d \pmod N . \tag{8}$$

Here d and N are parameters describing the generator. Marsaglia[16] discusses the performance of a generator in the form

$$x_n = x_{n-1} \times x_{n-2} \pmod{2^{32}} . \tag{9}$$

A more general form of this generator called Multiplicative Lagged-Fibonacci Generator (MLFG) is mentioned by Mascagni and Srivinasan[17] as

$$x_n = x_{n-j} \times x_{n-k} \pmod{2^m} , \qquad j < k . \tag{10}$$

Yet another example is the compound cubic congruential pseudorandom numbers[18] generated by the following formula:

$$y_{n+1}^{(i)} = a_i b_i^{-2} (y_n^{(i)} - c_i)^3 + b_i + c_i \pmod{p_i} . \tag{11}$$

Using these facts, we demonstrated the power of multiplication operation on improving a poor generator by combining with π. We therefore have implemented an integer multiplication before Mod operation and divided with the appropriate modulus. This gave very satisfactory results for the multiplicative case. Apart from generators considered by Tu and Fischbach, we have investigated the following very poor quality generators: Weyl sequence with $X_n = n\sqrt{2}$ (Mod 1), multiplicative congruential generators using $a = 732$ and 16 374 for $M = 32\,749$. This small modulus is chosen in order to demonstrate effectively the influence of coarse lattice structure on the output numbers. Our experiments on large moduli showed that in GRIP test, higher resolution of generated numbers can even hide the deficiency of generators having extremely bad lattice structure such as $X_n = 7X_{n-1}$ (Mod $2^{31} - 1$). Therefore for Tables 5 and 6 of Tu and Fischbach,[4] we examined the following set of generators and presented the test results in Tables 2–5:

(1) LCG1: The multiplicative random number generator $X_n = 16\,807X_{n-1}$ (Mod $2^{31} - 1$).
(2) F55a: The lagged Fibonacci generator using $X_n = (X_{n-55} + X_{n-24})$ (Mod 2^{31}).

(3) R31: Generalized feedback shift register (GFSR) generator using $X_n = X_{n-31} \oplus X_{n-3}$, where $\oplus$ is the bit-wise exclusive OR operation.
(4) NWS: The nested Weyl sequence $X_n = \{n\{n\alpha\}\}$, where $\{x\}$ is the fractional part of x.
(5) SNWS: The shuffled nested Weyl sequence generator $X_n = \{s_n\{s_n\alpha\}\}$, where $s_n = M\{n\{n\alpha\}\} + 0.5$ and M is a large positive integer.
(6) Weyl: The Weyl sequence obtained by $X_n = \{n\alpha\}$, where α is an irrational number. In our case, we took $\alpha = 2^{1/2} - 1$.
(7) LCG-A: It is a very poor multiplicative congruential random number generator of the form $X_n = 732X_{n-1}$ (Mod 32 749). Modulus 32 749 has a total of 10 912 multipliers satisfying full period. 732 is one of the eight extremely degenerate multipliers in the sixth dimension having $S_6 = 0.2373$.
(8) LCG-B: This generator having the form $X_n = 16\,374X_{n-1}$ (Mod 32 749) is also very poor. It is one of the four multipliers having $S_3 = 0.0623$ in third dimension. The lattice structure in two-dimensional space is presented in Fig. 1. Spectral test results of LCG-A and LCG-B are given in Table 1.

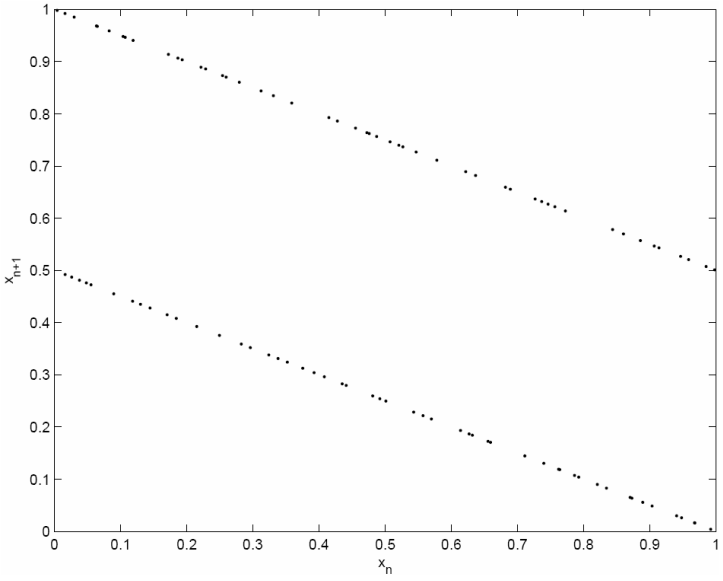

Fig. 1. The lattice distribution of X_n and X_{n+1} for the generator $X_n = 16\,374X_{n-1}$ mod 32 749.

Table 1. Figure of merits for LCG-A and LCG-B obtained by Spectral test.

Generator	S_2	S_3	S_4	S_5	S_6
LLG-A	0.9351	0.6319	0.5797	0.8435	0.2373
LCG-B	0.0115	0.0623	0.1398	0.2271	0.3063

Table 2. Computed results for $\langle \mathbf{r}_{12} \cdot \mathbf{r}_{23} \rangle_3$, where "Expected" means and variances are obtained from the exact distribution.

RNG	Mean	Variance	Error	RNG	Mean	Variance	Error
LCG1	−0.9958	1.2491	0.5563σ	LCG1, π, $\times$	−1.0044	1.2650	0.5828σ
F55a	−1.0035	1.2652	0.4636σ	F55a, π, $\times$	−0.9912	1.2576	1.1656σ
R31	−0.9848	1.2591	2.0133σ	R31, π, $\times$	−1.0058	1.2602	0.7682σ
NWS	−1.0114	1.9787	1.5100σ	NWS, π, $\times$	−1.0067	1.2758	0.8874σ
SNWS	−1.0054	1.2589	0.7152σ	SNWS, π, $\times$	−0.9897	1.2694	1.3643σ
Weyl	−0.7066	2.3380	38.8161σ	Weyl, π, $\times$	−0.9941	1.2689	0.7815σ
LCG-A	−1.0001	1.3026	0.0132σ	LCG-A, π, $\times$	−1.0059	1.2643	0.7815σ
LCG-B	−1.2661	2.8764	35.2456σ	LCG-B, π, $\times$	−1.0048	1.2808	0.6358σ
Expected	−1.0000	1.2667		Expected	−1.0000	1.2667	

For each entry in the table, 22 222 random observations are used.

Table 3. Computed results for $\langle \mathbf{r}_{12} \cdot \mathbf{r}_{23} \rangle_6$, where "Expected" means and variances are obtained from the exact distribution.

RNG	Mean	Variance	Error	RNG	Mean	Variance	Error
LCG1	−2.0010	2.4745	0.0662σ	LCG1, π, $\times$	−2.045	2.5244	0.2980σ
F55a	−2.0225	2.5407	1.4901σ	F55a, π, $\times$	−2.0310	2.5736	2.0530σ
R31	−1.9968	2.5718	0.2119σ	R31, π, $\times$	−2.0180	2.5099	1.1921σ
NWS	−2.0074	3.1273	0.4901σ	NWS, π, $\times$	−1.9978	2.5643	0.1457σ
SNWS	−1.9898	2.5792	0.6755σ	SNWS, π, $\times$	−1.9865	2.5753	0.8941σ
Weyl	−5.6523	0.5436	241.8784σ	Weyl, π, $\times$	1.9874	2.5037	0.8345σ
LCG-A	−2.0015	2.5933	0.0993σ	LCG-A, π, $\times$	−2.0045	2.5829	0.2890σ
LCG-B	−1.9500	3.4212	3.3113σ	LCG-B, π, $\times$	−1.9931	2.4983	0.4570σ
Expected	−2.0000	2.5333		Expected	−2.0000	2.5333	

For each entry in the table, 22 222 random observations are used.

We must stress here that the aim of this investigation is not to conduct a rigorous test of randomness for π. Because this was already done by several authors. For example, Marsaglia[19] subjected 10^9 digits of π to extensive and "difficult-to-pass" tests and noted that "it sailed through all of them" safely. For this reason by taking the first 1 000 000 decimal digits of $\pi - 3$, we demonstrated the performance of the above-mentioned generators and their output combined with π by multiplication operation $\times$. Random coordinates in n-dimensional cubes are calculated to 5 digits accuracy. By this way, various numbers of random observations are used for entries of each table. For example, Table 2 contains three dimensions each with three points. Therefore, the number of observations in this table is $1\,000\,000/(3 \times 3 \times 5) = 22\,222$. Similar calculation for Table 5 gives $1\,000\,000/(6 \times 4 \times 5) = 8333$ observations. The column called "Error" is calculated by dividing the absolute deviation of observed and expected means by the standard error obtained from the expected variance. For example, in Table 2, LCG1 has a mean value −0.9958. The error of this generator

Table 4. Computed results for $\langle(\mathbf{r}_{12}\cdot\mathbf{r}_{23})(\mathbf{r}_{34}\cdot\mathbf{r}_{41})\rangle_3$, where "Expected" means and variances are obtained from the exact distribution.

RNG	Mean	Variance	Error	RNG	Mean	Variance	Error
LCG1	1.3291	5.9660	0.2204σ	LCG1, π, $\times$	1.3336	6.1163	0.0141σ
F55a	1.3713	6.6830	1.9783σ	F55a, π, $\times$	1.3465	6.2060	0.6862σ
R31	1.3181	6.0108	0.7935σ	R31, π, $\times$	1.3318	6.0119	0.0797σ
NWS	2.0599	21.9184	37.8559σ	NWS, π, $\times$	1.3309	5.8380	0.1266σ
SNWS	1.3274	6.0507	0.3090σ	SNWS, π, $\times$	1.3390	6.3426	0.2954σ
Weyl	0.3125	0.0031	53.1875σ	Weyl, π, $\times$	1.3521	6.4011	0.9780σ
LCG-A	1.3420	8.2758	0.4533σ	LCG-A, π, $\times$	1.3366	6.1557	0.1719σ
LCG-B	2.3447	40.4105	52.6962σ	LCG-B, π, $\times$	1.3306	6.1026	0.1407σ
Expected	1.3333	6.1393		Expected	1.3333	6.1393	

For each entry in the table, 16 666 random observations are used.

Table 5. Computed results for $\langle(\mathbf{r}_{12}\cdot\mathbf{r}_{23})(\mathbf{r}_{34}\cdot\mathbf{r}_{41})\rangle_6$, where "Expected" means and variances are obtained from the exact distribution.

RNG	Mean	Variance	Error	RNG	Mean	Variance	Error
LCG1	4.6179	36.1011	0.7350σ	LCG1, π, $\times$	4.7149	37.0267	0.7269σ
F55a	4.7315	38.5338	0.9771σ	F55a, π, $\times$	4.6779	36.2887	0.1693σ
R31	4.6743	36.0077	0.1150σ	R31, π, $\times$	4.6614	35.5602	0.0794σ
NWS	4.6923	43.7916	0.3863σ	NWS, π, $\times$	4.6946	38.4139	0.4209σ
SNWS	4.6183	37.5326	0.7290σ	SNWS, π, $\times$	4.6866	36.8059	0.3004σ
Weyl	32.3188	55.0091	416.745σ	Weyl, π, $\times$	4.5352	34.9914	1.9814σ
LCG-A	4.7070	38.4749	0.6078σ	LCG-A, π, $\times$	4.5733	33.7631	1.4072σ
LCG-B	4.6951	53.6207	0.4285σ	LCG-B, π, $\times$	4.6682	36.6084	0.0231σ
Expected	4.6667	36.6874		Expected	4.6667	36.6874	

For each entry in the table, 8333 random observations are used.

is

$$\frac{|-0.9958-(-1.0000)|}{\sqrt{1.2667/22222}} = 0.5563\,.$$

Results can be summarized as follows:

(1) The combination with π using $\times$ operation improves the performance of generators substantially. Although there are occasional cases exhibiting an increased error, repetition with different seeds indicated that these are within the limits of chance variation.

(2) Weyl sequence fails the GRIP tests drastically in $\langle(\mathbf{r}_{12}\cdot\mathbf{r}_{23})(\mathbf{r}_{34}\cdot\mathbf{r}_{41})\rangle_n$ and $\langle\mathbf{r}_{12}\cdot\mathbf{r}_{23}\rangle_n$ for both $n=3$ and 6 (Tables 2–5). The errors of $n=6$ are larger than errors of $n=3$. As can be seen in Figs. 2 and 3, the first difference $X_{i+1}-X_i$ of Weyl series is a dichotomous constant value as $X_{i+1}-X_i=\alpha$ or $\alpha-1$, GRIP test effectively finds this deficiency by employing the difference of consecutive elements. It is remarkable that combining with π eliminates this defect.

(3) As explained by Holian *et al.*,[20] the nested Weyl sequence is characterized by constant second differences $(X_{i+1}-X_i)-(X_i-X_{i-1})$. Here we have $X_{i+1}-$

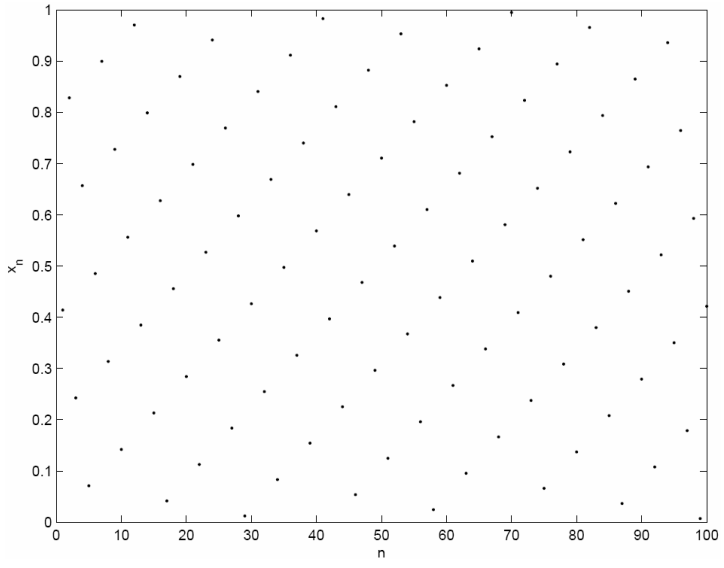

Fig. 2. The first 100 elements of the Weyl sequence.

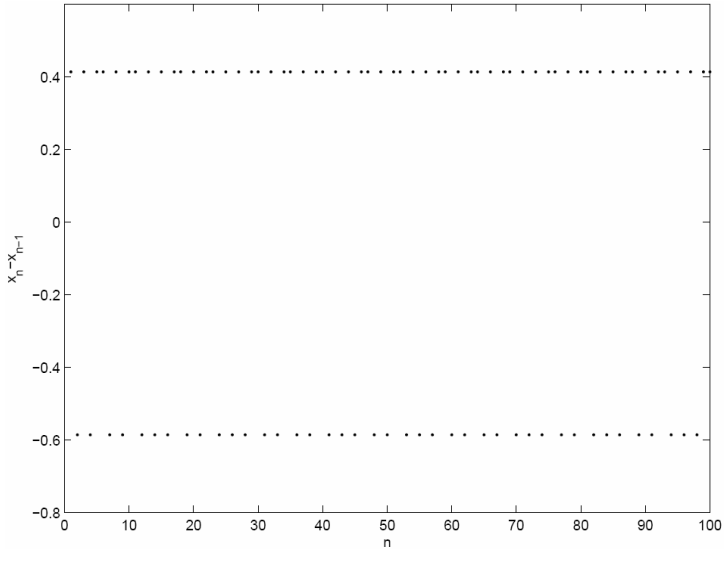

Fig. 3. 100 first-order differences $X_{i+1} - X_i$ of the Weyl sequence.

$2X_i + X_{i-1} = 2\alpha - j$, where $j = -1, 0, +1, +2$ as shown in Figs. 4 and 5. (In work by Holian *et al.*,[20] the expression $X_{i+1} - 2X_i - X_{i-1}$ for second difference is wrong.) The GRIP test effectively demonstrates this defect. However, the combination with π using $\times$ operation improves the performance.

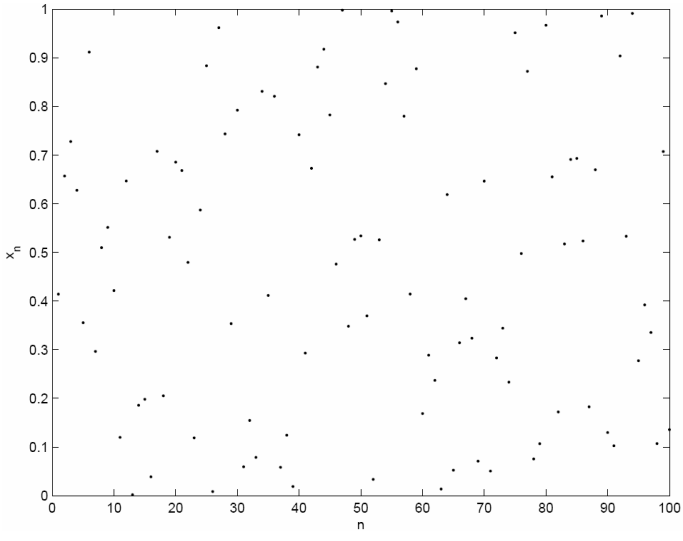

Fig. 4. The first 100 elements of NWS.

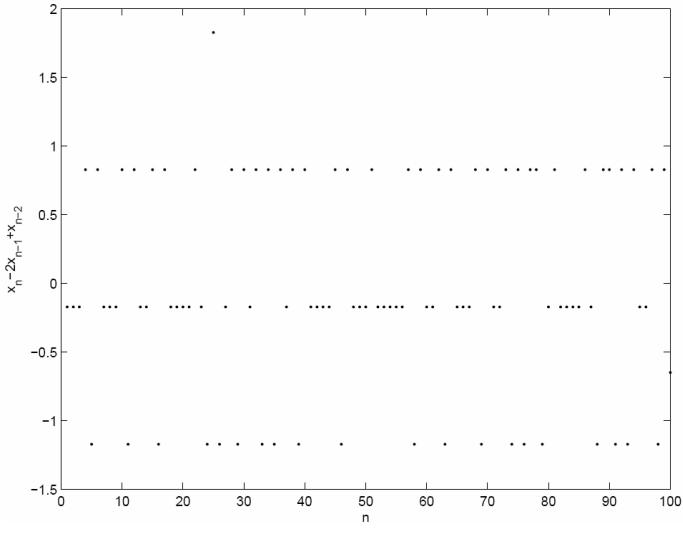

Fig. 5. 100 second-order differences $X_{i+1} - 2X_i + X_{i-1}$ for NWS.

(4) LCG-B having bad lattice structure in low dimensions gives unsatisfactory results when tested alone. However, its combination with π improves the performance.

(5) Using sample estimates of μ_4 and μ_2, and noting that the sample variance S^2 has a mean value of σ^2 and variance $(\mu_4 - \mu_2^2)/n$ in large samples, our standard normal test

$$z = \frac{S^2 - \sigma^2}{\sqrt{(\mu_4 - \mu_2^2)/n}} \tag{12}$$

revealed that the variances of generators NWS, Weyl, LCG-A, and LCG-B significantly deviate from the parametric values. Multiplicative combining with π improves this situation.

4. Some Additional Remarks

4.1. *Objects and derived distributions*

GRIP family of tests can be applied to geometric objects with different shapes such as spheres, ellipsoids, sphere surfaces, or n-cubes. In spherical surfaces, it is possible to choose Euclidean or geodesic distances. Moreover, one can choose probability distributions other than uniform. In some studies,[1,3] Tu and Fischbach used spherical objects. However, objects other than n-cubes are not suitable in statistical testing for two reasons: These objects ignore some of the data. For example, the spherical objects choose only the points less than a specific distance to the origin. The omitted portion of points is not negligible. For the circle in two-dimensional space, a ratio of $\pi/4 = 78.5\%$ of data is tested. In three-dimensional space, the sphere occupies only 52.4% of the volume. This means that for the circle 21.5% and for the sphere 47.6% of the data are not analyzed. According to our simulation with the congruential random number generator with multiplier 41 358 and modulus $2^{31} - 1$, in six-dimensional space 91.9% of the data is wasted.

The wasted proportion rises to 99.4% in nine dimensions. This situation will have several adverse effects on the simulation results. First of all, it will diminish the efficiency of the test. In evaluating the difference in performance of tests in various dimensions, this effect must be kept in mind. A more dangerous situation may arise if the tested distribution has heterogeneous concentration of points in the space. Especially if points exhibit an aggregation near the corners of the hypercube or within the spherical object, omission of points beyond the boundary of n-sphere will distort the test results. In commenting on results to larger mean values for the case of $n = 9$ in their Table 2, Tu and Fischbach[3] argue that "One interpretation may be that $\langle(\mathbf{r}_{12} \cdot \mathbf{r}_{23})(\mathbf{r}_{34} \cdot \mathbf{r}_{41})\rangle_9$ is a more sensitive and dedicated computational test for investigating random number generators than other GRIP tests." This argument needs a more convincing theoretical support. However, one plausible explanation for this fact may be that the selection process is eliminating more than 99% of the data and the remaining points behave in a biased manner. The test is based on a very small number of points that are concentrated in the central region of the n-cube. This bias may be due also to insufficient accuracy of real number representations. Another important point is the actual sample size N. Because the apparent sample size may seem to be 10^6 or 10^8 but the actual sample size is only the number of points falling within the geometric object studied. Necessary adjustments must be provided in the computer programs.

Testing derived distributions is not very informative for the quality of a random number generator. Because deficiencies are inherited from the generator producing the uniform numbers. For example, the lattice structure of congruential random number generators are reflected to normal variables in the form of spirals known as Neave effect in the output of Box–Muller transformation. Similar pathological phenomena are observed in Tausworth and GFSR sequences. The crucial element is the quality of uniform distribution. If the starting point is free of defect, it is easy to choose reliable techniques for obtaining nonuniform random variables. More research is needed to find examples where it is not possible to spot defects in the uniform distribution but the GRIP test can reveal it in a derived distribution.

4.2. *Choice of the generators*

One of the primary objectives of Tu and Fischbach[1,3] was to demonstrate the performance of GRIP test on various random number generators. In their first study, three generators are tested. In later studies, this number is raised to 10 and 25, respectively. In their last study,[4] they compare π with 30 other random number generators available and common in applied literature. A greater care should be spent to this choice because there are several other generators in literature of high quality and popular usage. Some information on generators studied in this paper is incorrect. For example, the standard UNIX library function DRAND48 presented as LGC3 is not listed among seven 48-bit generators studied by the referred work of Fishman. Moreover, this generator is not a multiplicative as stated by authors, but it is a mixed congruential generator. There is a serious problem for modulus and periods of LCG4 and LCG5 because $2^{63} - 1$ is not a prime. The largest prime for 64-bit word-size is $2^{63} - 25$. Therefore, LCG4 and LCG5 do not have full period.

4.3. *Spurious precision and waste of data*

Tu and Fischbach[2] present their data in 9-digits accuracy. Later this accuracy is extended to 10 digits.[3,4] These accuracies cannot be realized if they did not use double-precision real variable definitions. In the Appendix of Ref. 4, authors listed a sample Fortran 90/95 program for calculating GRIP test. They claim to be using a 10-digit string for each coordinate point to achieve a precision sufficient for their testing purposes. It is not obvious why 10-digit precision is required. However, even if this is the case, the compiler accuracy is not adequate to realize it. On the contrary, unfortunately about 30% of data is wasted by this approach because the precision of FORTRAN real numbers is not enough to represent these inputs. In addition, not all decimal numbers have an exact binary representation in a computer because the mantissa is always truncated at some stage. According to American National Standards Institute (ANSI) and Institute of Electrical and Electronic Engineers (IEEE) 754/1985 standard, a single-precision word-size of 32-bit register assigns one bit for the sign, 8 bits for the exponent and only 23 bits for the mantissa of a real number. The numerical precision is limited to sixth or seventh

decimal digit since the least significant digit is given by $1/2^{23} \approx 10^{-7}$. The same problem applies to entries of Tables 1–6 of Ref. 4 and Tables 1–4 of Ref. 3 where data are reported with implications of precision which are not justified. Another precision problem is caused by the misleading FORTRAN program in the Appendix of Ref. 4. In order to obtain one million random points in three dimensions, we need 90 million decimal digits. The program gives only one-digit numbers. Therefore, the array pi should have been declared as pi(90000000) and before assigning data to vector *random*, 10-digit strings should have been formed.

5. Conclusions

We discussed a newly developed randomness test method, GRIP and its application to decimal digits of π. We stress that a sound definition of significance test is required. For this purpose, the best way is to determine the exact distribution function of the test statistic and its percentiles. The work of Marsaglia proves that this distribution is extremely complicated. Another approach may be the usage of parametric values of means and variances. In that case, hypothesis testing may be accomplished by using normal approximations. We give parametric values for several cases. More reliable test results can be obtained by dividing the data into subsets and employing goodness of fit tests such as Anderson–Darling, Kolmogorov–Smirnov or Cramer–von Mises. We also stress the need for determining the meaning of this test rigorously and classes of defects particularly diagnosed by it. According to our findings, GRIP is able to detect shortcomings of random numbers having unwanted regularities in their first- or second-order differences, such as Weyl and nested Weyl sequences. There is a need to compare GRIP with conventional test methods and explain similarities. For example, GRIP is successful in diagnosing deficiency of short-period generators failing in spectral test because of bad lattice structures. We also discussed in details some shortcomings of the work about the randomness of π. By appropriate implementation, we proved that π is able to correct bad outputs of inadequate random numbers by multiplicative combining method. We made some more comments and suggested some corrections on the works of Tu and Fischbach.

Acknowledgments

I would like to thank my son Dr Tevfik Metin Sezgin, Assistant Professor at Koç University Computer Science Department, for his useful discussion concerning certain equations and contributions for the LATEX version of the paper.

Appendix A. Geometric Random Inner Product Test Program Examples

```
!*****************************************************************
!   Geometric Inner Product Test program
```

```
!   In this program N is the size of the uniform data to be tested.
!   Users can define it according to their data size.
!*********************************************************************
       parameter (N=10000)
       dimension  GRIP(6,8),  xrandom(N)
! ***********************************************
!        Dimension = 3, Points=3
! ***********************************************
       sgrip=0
       ssgrip=0
       k=0
       krow=3
       kcol=3
       Loop=N/( krow*kcol)
       do L=1,Loop
          gr=0
          do i=1,krow
             do j=1,kcol
                k=k+1
                grip(i,j)=2*xrandom(k)-1.0
             end do
              gr=gr+(grip(i,2)-grip(i,1))*(grip(i,3)-grip(i,2))
          end do
          sgrip=sgrip+gr
          ssgrip=ssgrip+gr*gr
       end do
       grmean=sgrip/loop
       grvar=(ssgrip-sgrip*grmean)/(Loop-1)
       zgrip=(grmean+1)/sqrt(1.26667/Loop)
       write (2,1) "Geometric Random Inner Product Test            "
       write (2,2) "Dimension=",kcol, "Points=",krow, "Z=",zgrip
       write (2,3) "Sample size (number of simulations)= ",loop
       write (2,4) "Mean value=", grmean, "Variance=", grvar
 ! ***********************************************
 !        Dimension = 4, Points=3
 ! ***********************************************
       sgrip=0
       ssgrip=0
       k=0
       krow=3
       kcol=4
       Loop=n1/(mbits*krow*kcol)
       do L=1,Loop
         gr=0
         do i=1,krow
            do j=1,kcol
               k=k+1
               grip(i,j)=2*xrandom(k)-1.0
            end do
         end do
         f1=(grip(1,2)-grip(1,1))*(grip(1,3)-grip(1,2))+
      *     (grip(2,2)-grip(2,1))*(grip(2,3)-grip(2,2))+
      *     (grip(3,2)-grip(3,1))*(grip(3,3)-grip(3,2))
         f2=(grip(1,4)-grip(1,3))*(grip(1,1)-grip(1,4))+
      *     (grip(2,4)-grip(2,3))*(grip(2,1)-grip(2,4))+
```

```
*     (grip(3,4)-grip(3,3))*(grip(3,1)-grip(3,4))
      gr=f1*f2
      sgrip=sgrip+gr
      ssgrip=ssgrip+gr*gr
     end do
     grmean=sgrip/loop
     grvar=(ssgrip-sgrip*grmean)/(loop-1)
     zgrip=(grmean-1.3333333)/sqrt(6.1393/loop)
     write (2,1) "                                        "
     write (2,2) "Dimension=",kcol, "  Points=",krow,"  Z=",zgrip
     write (2,3) "Sample size (number of simulations)= ",loop
     write (2,4) "Mean value=", grmean, "Variance=", grvar
1    format(/, a47,/)
2    format(a10,i2,a8,i2,a4,f6.3)
3    format(a37,i4)
4    format(a11,f6.3,a14,f6.3)
```


References

1. S. J. Tu and E. Fischbach, *J. Phys. A: Math. Gen.* **35**, 6557 (2002).
2. S. J. Tu and E. Fischbach, **1** (2002), arXiv:physics/0209032.
3. S. J. Tu and E. Fischbach, *Phys. Rev. E* **67**, 016113-1 (2003).
4. S. J. Tu and E. Fischbach, *Int. J. Mod. Phys. C* **16**, 281 (2005).
5. R. K. Pathria, *Math. Comput.* **16**, 79 (1962).
6. B. R. Johnson and D. J. Leeming, *Sankhya* **52**, 183 (1990).
7. Y. Dodge, *Int. Statist. Rev.* **3**, 329 (1996).
8. T. Jaditz, *Amer. Statist.* **54**, 12 (2000).
9. M. G. Kendall and A. Stuart, *The Advanced Theory of Statistics*, Vol. 1, 3rd edn. (Charles Griffin & Co., London, 1969), p. 277.
10. D. E. Knuth, *The Art of Computer Programming*, Vol. 2, 3rd edn. (Reading MA, Addison Wesley, 1998), p. 41.
11. M. A. Stephens, in *Goodness of Fit Techniques*, Statistics Textbooks and Monographs, Vol. 68, eds. R. B. D'Agustino and M. A. Stephens (Marcel Dekker, 1986), pp. 97–194.
12. G. Marsaglia, *InterStat* **1** (2006).
13. L. Y. Deng and E. O. George, *Annu. Inst. Statist. Math.* **44**, 379 (1992).
14. L. Devroye, *Non-Uniform Random Variable Generation* (Springer Verlag, New York, 1986), p. 24.
15. J. C. Lagarias, *Statist. Sci.* **8**, 31 (1993).
16. G. Marsaglia, *Proc. Symposia in Applied Mathematics* **46**, 73 (1992).
17. M. Mascagni and A. Srivinasan, *ACM Trans. Math. Software* **26**, 436 (2000).
18. J. Eichenauer-Herrmann and E. Herrmann, *Computing* **59**, 85 (1997).
19. G. Marsaglia, *InterStat* **5** (2005).
20. B. L. Holian, O. E. Percus, T. T. Warnock and P. A. Whitlock, *Phys. Rev. E* **50**, 1607 (1994).